Computational Geosciences manuscript No.
(will be inserted by the editor)

Well Placement Optimization with the Covariance Matrix Adaptation Evolution Strategy and Meta-Models

Zyed Bouzarkouna · Didier Yu Ding · Anne Auger

Received: date / Accepted: date

Abstract The amount of hydrocarbon recovered can be considerably increased by finding optimal placement of non-conventional wells. For that purpose, the use of optimization algorithms, where the objective function is evaluated using a reservoir simulator, is needed. Furthermore, for complex reservoir geologies with high heterogeneities, the optimization problem requires algorithms able to cope with the non regularity of the objective function.

In this paper, we propose an optimization methodology for determining optimal well locations and trajectories based on the Covariance Matrix Adaptation - Evolution Strategy (CMA-ES) which is recognized as one of the most powerful derivative-free optimizers for continuous optimization. In addition, to improve the optimization procedure two new techniques are proposed:

- Adaptive penalization with rejection in order to handle well placement constraints;
- Incorporation of a meta-model, based on locally weighted regression, into CMA-ES, using an approximate stochastic ranking procedure, in order to reduce the number of reservoir simulations required to evaluate the objective function.

The approach is applied to the PUNQ-S3 case and compared with a Genetic Algorithm (GA) incorporating the Genocop III technique for handling constraints. To allow a fair comparison, both algorithms are used without parameter tuning on the problem, standard settings are used for the GA and default settings for CMA-ES. It is shown that our new approach outperforms the genetic algorithm: it leads in general to both a higher net present value and a significant reduction in the number of reservoir simulations needed to reach a good well configuration. Moreover, coupling CMA-ES with a meta-model leads to further improvement, which was around 20% for the synthetic case in this study.

Keywords Well placement · Field development optimization · CMA-ES · Evolution strategies · Meta-models

Z. Bouzarkouna · D. Ding
IFP Energies nouvelles
1 & 4 avenue de Bois-Préau
92852 Rueil-Malmaison cedex, France
E-mail: zyed.bouzarkouna@ifpen.fr

D. Ding
E-mail: d-yu.ding@ifpen.fr

Z. Bouzarkouna · A. Auger
TAO Team - INRIA Saclay-Île-de-France
LRI - Paris-Sud University
91405 Orsay cedex, France

A. Auger
E-mail: anne.auger@inria.fr

1 Introduction

A well placement decision affects the hydrocarbon recovery and thus the asset value of a project. In general, such a decision is difficult to make since optimal placement depends on a large number of parameters such as reservoir heterogeneities, faults and fluids in place. Moreover, dealing with complex well configurations, e.g., non-conventional wells, implies additional challenges such as the concentration of investment and the well intervention difficulty.

A well placement decision can be formulated as an optimization problem: the objective function optimized evaluates the economics of the project; the parameters thought encode the position of the different wells. If the number of wells to be placed and their type (injector or producer) are fixed, the parameters encoding the well positions (that include locations and trajectories) are

real numbers and the objective function f maps a subset of $\mathbb{R}^n$ where n, the number of parameters, equals the number of parameters needed to encode a well position times the number of wells that need to be placed.

However, the optimization task is challenging as the objective function, e.g., the net present value (NPV) is in general *multi-modal*, i.e., with multiple local optima, *non-convex* and *non-smooth*. An illustration can be found in [47] where the NPV of a single vertical well placement is sampled to construct the objective function surface. The surface is shown to be highly non-smooth and to contain several local optima. In this illustration, the problem dimension equals two and it has thus been possible to sample all the points from a fine grid spanning regularly the search space. However, this becomes impossible for problem dimensions larger than 3 as the number of points, to keep a fine discretization, would need to grow exponentially in the search space dimension (this is referred as *curse of dimensionality*) rendering the search task difficult. In addition, the problem is *costly*: a single function evaluation requires one reservoir simulation which is often very demanding in CPU time (several minutes to several hours).

Furthermore, constraints are imposed to guarantee the physical feasibility of the solution wells and avoid either undrillable wells (e.g., very long wells) or wells that violate common engineering practices (e.g., wells outside the reservoir). Therefore, a constraint optimization problem needs to be handled.

Well placement optimization is a recent area of research that is gaining growing interest. Different methodologies have been investigated. On the one hand, approaches based on stochastic search algorithms were used, where minimal assumptions on the problem are needed and that are thus more robust than deterministic methods when dealing with rugged problems: simulated annealing (SA) [8], particle swarm optimization (PSO) [47], genetic algorithm (GA) [44,18, 45,13], simultaneous perturbation stochastic algorithm (SPSA) [5,6]. On the other hand, deterministic optimization methods were also used. Gradient-based algorithms were mostly used, where the underlying model of the function needs to be smooth enough and in which adjoint methods were used for computing the gradients [28,50,23,53,56].

Stochastic algorithms have been combined with proxy-models –called also surrogates, meta-models or response surface models– and deterministic approaches: GA with a polytope algorithm and kriging [24,25], GA with a polytope algorithm, kriging and neural networks [26], GA with neural networks, a hill climber and a near-well upscaling technique [55]. Results show that a hybrid stochastic algorithm converges in general to a reasonable solution with a reduced number of evaluations compared to a pure stochastic algorithm.

The approaches in [24–26,55] build at each iteration a proxy-model, determine its maximum and include the location of this maximum in the population (replacing the worse individual) if it is better than the best individual of the current population. In [3], a GA is defined, in which at each iteration, only a predefined percentage of the individuals, chosen according to a set of scenario attributes, is simulated. The objective function of the non-simulated points is estimated using a statistical proxy based on cluster analysis.

Other approaches, called proxy-modeling or meta-modeling in the literature, are based on replacing the true objective function by a proxy-model and applying the optimization techniques to the proxy, and not to the true objective function. Proxy-models may include least squares and kriging [48], radial basis functions [19], quality maps [15,46]. Although proxy-modeling is an efficient way to have an approach with a reduced number of reservoir simulations, its application, with increasing complexity of the solution space, is not recommended [57].

In this paper, we propose a new methodology for well location and trajectory optimization based on the population based stochastic search algorithm called Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [33]. CMA-ES is both a fast and robust local search algorithm, exhibiting linear convergence on wide classes of functions and a global search algorithm when playing with restart and increase of population size. Intensive benchmarking of several derivative free algorithms have established that CMA-ES is one of the most efficient method for dealing with difficult numerical optimization problems [29]. CMA-ES has also been applied to real-world problems [7,16,39,40]. CMA-ES, in contrast to most other evolutionary algorithms, is a quasi parameter-free algorithm[1].

To the best of our knowledge, only two studies have applied CMA-ES in the petroleum industry: a characterization of fracture conductivities from well tests inversion [12] and a well placement optimization but with respect to simple attributes (e.g., productivity indices) [17].

In this work we apply CMA-ES to the well placement problem using adaptive penalization with rejection to handle constraints. Because genetic algorithms are quite often the method of choice in petroleum industry, we first show the improvement of applying CMA-ES

[1] Only the population size is suggested to be adjusted by the user in order to account for the ruggedness of the objective function landscape.

over a GA on the synthetic benchmark reservoir case PUNQ-S3 [21]. In addition, because a reservoir simulation and thus the objective function is expensive, we couple CMA-ES with locally weighted quadratic meta-models aiming at saving a number of evaluations by building a model of the problem. We validate the approach on the PUNQ-S3 case.

This paper is structured as follows. Section 2 describes the problem formulation. Section 3 gives an overview of the optimization algorithm CMA-ES and presents constraint handling with adaptive penalization and rejection. In Section 4, CMA-ES is compared to a genetic algorithm on a synthetic reservoir case to show the contribution of the proposed optimization method. In Section 5, the reduction of the number of reservoir simulations is addressed by coupling CMA-ES with meta-models. Finally, in Section 6, the contribution of the whole methodology, i.e., CMA-ES with meta-models is demonstrated on a number of well location and trajectory optimization problems (with unilateral and multilateral wells).

2 The well placement optimization problem formulation

In this section, we describe the well placement optimization problem and explain the parameterization of the wells.

2.1 Objective function

The objective function associated with a well placement problem often evaluates the economic model of the decision and takes into account different costs such as prices of oil and gas, costs of drilling and costs of injection and production of water. Another alternative is to use the cumulative oil production or the barrel of oil equivalent (BOE). In this paper, the objective function considered is the net present value NPV [55] defined by

$$\mathrm{NPV} = \sum_{n=0}^{Y} \left(\frac{1}{(1+\mathrm{APR})^n} \begin{bmatrix} Q_{n,o} \\ Q_{n,g} \\ Q_{n,w_a} \end{bmatrix}^T \begin{bmatrix} C_{n,o} \\ C_{n,g} \\ C_{n,w_a} \end{bmatrix} \right) - C_d \ , \quad (1)$$

where $Q_{n,p}$ is the field production of phase p in the period n and $C_{n,p}$ is the profit or loss associated to this production. A phase p is either oil, gas or water denoted respectively by o, g, w_a. The annual percentage rate is denoted APR. The integer Y is the number of discount periods. The constant C_d is the drilling and completing cost for all wells approximated by the following equation

$$C_d = \sum_{w=0}^{\mathcal{N}_\mathrm{w}} \left(\sum_{k=0}^{\mathcal{N}_\mathrm{lat}} \left[\mathrm{A}.d_w.\ln(l_w).l_w \right]_{k,w} \right) + \sum_{m=1}^{\mathcal{N}_\mathrm{jun}} \left[C_{jun} \right]_m \ , \quad (2)$$

based on the approximate formula used in [55]. In Eq. 2, $k = 0$ represents the mainbore, $k > 0$ represents the laterals, l_w is the length of the lateral (in ft), d_w is the diameter of the mainbore (in ft), $\mathcal{N}_\mathrm{w}$ is the number of wells drilled and A is a constant specific to the considered field. The constant C_{jun} is the cost of milling the junction. The constants in Eqs. 1 and 2 defined for the examples hereafter are given in Table 1.

Table 1 Constants used to define the net present value (NPV).

Constant	Value	
$C_{n,o}$	60	\$ / barrel
C_{n,w_a}	-4	\$ / barrel
$C_{n,g}$	0	\$ / barrel
APR	0.1	
A	1000	
d_w	0.1	m
C_{jun}	10^5	\$

The field productions $Q_{n,p}$ for phases $p = o, g, w_a$ are obtained for a given well configuration using a reservoir simulator which represents *the time consuming part* in the computation of the NPV objective function.

To guarantee a feasible solution, different constraints defining the feasible optimization domain were implemented. The constraints are as follows:

- maximum length of wells: $l_w < L_{\max}$, for each well w to be placed;
- all wells must be inside the reservoir grid: $l_w = l_{\mathrm{inside}}$, for each well w to be placed, where l_{inside} is the length of the well w inside the reservoir grid.

2.2 Well parameterization

In our approach, we want to be able to handle different possible configurations of multilateral wells. An illustrative scheme is given in Fig. 1. The terminology used to define each part of a multilateral well follows the terminology used in [34]. In general, a lateral can be defined by a line connecting two points. The mainbore is defined through the trajectory of its contiguous completed segments. Hence, we define a sequence of points where a deviation occurs $(\mathrm{P_{d,i}})_{0 \le \mathrm{i} \le \mathcal{N}_\mathrm{s}}$ where $\mathcal{N}_\mathrm{s}$ is the number of segments. The starting point $\mathrm{P_{d,0}} = \mathrm{P_0}$ of the mainbore called the heel is represented by its Cartesian coordinates (x_0, y_0, z_0). Other intermediate points $(\mathrm{P_{d,i}})_{1 \le \mathrm{i} \le \mathcal{N}_\mathrm{s}-1}$ and the ending point $\mathrm{P_{d,\mathcal{N}_s}}$ called the

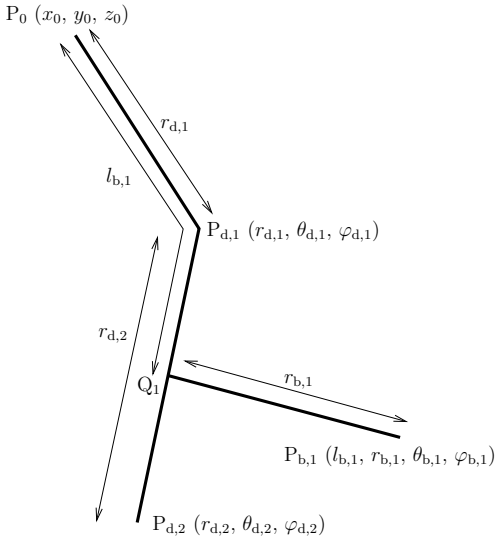

Fig. 1 An example of a single multilateral well parameterization with two segments ($\mathcal{N}_\mathrm{s} = 2$) and one branch ($\mathcal{N}_\mathrm{b} = 1$).

toe are represented by their corresponding spherical coordinate system ($r_{\mathrm{d,i}}$, $\theta_{\mathrm{d,i}}$, $\varphi_{\mathrm{d,i}}$) with respect to the basis ($\mathrm{P}_{\mathrm{d,i-1}}$, $\mathbf{u}^r_{\mathrm{d,i}}$, $\mathbf{u}^\theta_{\mathrm{d,i}}$, $\mathbf{u}^\varphi_{\mathrm{d,i}}$). We use spherical coordinates because they allow for straightforward control of the well lengths by imposing a box constraint whereas it would need to be handled by imposing a non linear constraint with Cartesian coordinates.

The wells are parameterized in a way to handle a number $\mathcal{N}_\mathrm{b}$ of branches and/or laterals as well.

The branch or lateral $j \in [1, \cdots, \mathcal{N}_\mathrm{b}]$ is defined by locating its ending point $\mathrm{P}_{\mathrm{b,j}}$ ($l_{\mathrm{b,j}}$, $r_{\mathrm{b,j}}$, $\theta_{\mathrm{b,j}}$, $\varphi_{\mathrm{b,j}}$) where $(r_{\mathrm{b,j}}, \theta_{\mathrm{b,j}}, \varphi_{\mathrm{b,j}})_{1\leq \mathrm{j} \leq \mathcal{N}_\mathrm{b}}$ represents the spherical coordinates of $\mathrm{P}_{\mathrm{b,j}}$ with respect to the basis ($\mathrm{Q_j}$, $\mathbf{u}^r_{\mathrm{b,j}}$, $\mathbf{u}^\theta_{\mathrm{b,j}}$, $\mathbf{u}^\varphi_{\mathrm{b,j}}$), $\mathrm{Q_j}$ is the starting point of the branch or the lateral j, and $l_{\mathrm{b,j}}$ is the distance along the well between P_0 and $\mathrm{Q_j}$.

The dimension D_w of the representation of a well denoted by w is as follows:

$$D_w = 3\ (1 + \mathcal{N}^w_\mathrm{s}) + 4\ \mathcal{N}^w_\mathrm{b}\ . \tag{3}$$

Hence, the dimension D of the problem of placing $\mathcal{N}_\mathrm{w}$ wells $(w_k)_{k=1,\cdots,\mathcal{N}_\mathrm{w}}$ is

$$D = \sum_{k=1}^{\mathcal{N}_\mathrm{w}} D_{w_k}\ . \tag{4}$$

An example of a single well parameterization is shown in Fig. 1. In this example, $\mathcal{N}_\mathrm{s}$ is equal to two and $\mathcal{N}_\mathrm{b}$ is equal to one. The mainbore is then represented by three points P_0 and $(\mathrm{P}_{\mathrm{d,i}})_{1\leq \mathrm{i} \leq 2}$. The branch is represented by one point $\mathrm{P}_{\mathrm{b,1}}$. The corresponding dimension of the optimization problem is 13.

3 The Covariance Matrix Adaptation - Evolution Strategy (CMA-ES)

In this section, we introduce the CMA-ES algorithm. The objective function to be optimized is denoted $f : \mathbb{R}^n \to \mathbb{R}$. We propose also an adaptive penalization with rejection technique to handle constraints.

3.1 CMA-ES

CMA-ES [33,30] is an iterative stochastic optimization algorithm where at each iteration, a population of candidate solutions is sampled. In contrast to the classical presentation of population based stochastic search algorithms (like genetic algorithms) where the different steps of the algorithms are described in terms of operators acting on the population (crossover, mutation), the natural algorithm template for CMA-ES translates the evolution of the probability distribution used to sample points at each iteration. Indeed, the algorithm loops over the following steps:

1. sample a population of λ candidate solutions (points of $\mathbb{R}^n$)
2. evaluate the λ candidate solutions on f
3. adapt the sampling distribution (using the feedback from f obtained at step 2.)

We see that this general template depends on a probability distribution (sampling distribution) and on the update of this probability distribution. The sampling distribution in CMA-ES is a *multivariate normal distribution*. In the next paragraphs we will give more insights on multivariate normal distributions and their geometrical interpretation and then explain how its update is performed at each iteration within CMA-ES.

Multivariate normal distributions A random vector of $\mathbb{R}^n$ distributed according to a multivariate normal distribution is usually denoted by $\mathcal{N}(\mathbf{m}, \mathbf{C})$ where $\mathbf{m}$ is a vector of $\mathbb{R}^n$ and $\mathbf{C}$ an $n \times n$ symmetric positive definite matrix corresponding to the covariance matrix of the random vector. The set of parameters $(\mathbf{m}, \mathbf{C})$ entirely determines the random vector. Fig. 2 gives the geometric interpretation of a random vector $\mathcal{N}(\mathbf{m}, \mathbf{C})$ in two dimensions. We visualize that $\mathbf{m}$ is the symmetry center of the distribution and that isodensity lines are ellipsoid centered in $\mathbf{m}$ with main axes corresponding to eigenvectors of $\mathbf{C}$ and lengths determined by the square roots of the eigenvalues of $\mathbf{C}$. Fig. 2 depicts also points sampled according to a multivariate normal distribution. As expected, the spread of the points follows the isodensity lines. A useful relation is $\mathbf{m} + \mathcal{N}(\mathbf{0}, \mathbf{C}) = \mathcal{N}(\mathbf{m}, \mathbf{C})$ that interprets $\mathbf{m}$ as the displacement from the origin $\mathbf{0}$.

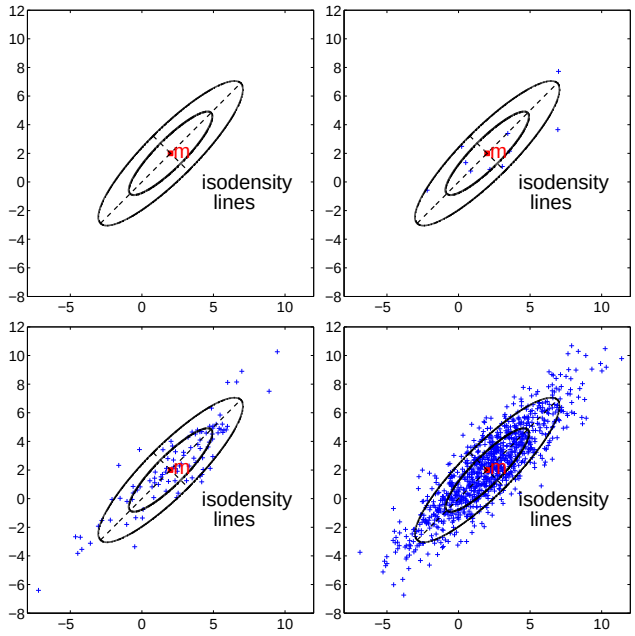

Fig. 2 Geometrical representation of a 2-dimensional multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$ where $\mathbf{m} = (2, 2)^T$ and the covariance matrix $\mathbf{C}$ admits $\frac{1}{\sqrt{2}}(1, 1)$ and $\frac{1}{\sqrt{2}}(-1, 1)$ as normalized eigenvectors with respective eigenvalues 16 and 1. Depicted on each plot is the mean vector $\mathbf{m}$ and the ellipsoid isodentity lines defined as $(\mathbf{x}-\mathbf{m})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{m}) = c$ where the constant c equals 1 (inner line) and 3 (outer line). The main axes of the (isodensity) ellipsoid are carried by eigenvectors of $\mathbf{C}$. The half lengths of the axes of the unit isodensity lines ($(\mathbf{x}-\mathbf{m})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{m}) = 1$) are the square roots of the eigenvalues of $\mathbf{C}$.
Depicted on the 2nd, 3rd and 4th plots are samples among 10 (resp. 100 and 1000) samples from $\mathcal{N}(\mathbf{m}, \mathbf{C})$ falling into the box plot $[-8, 12] \times [-8, 12]$.

In CMA-ES, the mean vector represents the favorite solution or best estimate of the optimum, and the covariance matrix $\mathbf{C}$ characterizing the geometric shape of the distribution defines where new solutions are sampled. Furthermore, an additional parameter is added, which is the step-size σ used as a global scaling factor for the covariance matrix. Overall, in step 1. for CMA-ES, points are sampled according to

$$\mathbf{m} + \sigma\mathcal{N}(\mathbf{0}, \mathbf{C}) \ .$$

The adaptation of $\mathbf{m}$ targets to find the best estimate of the optimum, the adaptation of $\mathbf{C}$ aims at learning the right coordinate system of the problem (rotation and scaling of the main axes) and the adaptation of σ aims at achieving fast convergence to an optimum and preventing premature convergence. We will now describe how the distribution is updated, that is how the parameters $\mathbf{m}$, σ and $\mathbf{C}$ are updated in step 3. of the template.

Update of mean vector, covariance matrix and step-size We adopt here some time-dependent notations. The iteration index is denoted g. Let $(\mathbf{m}^{(g)}, g \in \mathbb{N})$ be the sequence of mean vectors of the multivariate normal distribution generated by CMA-ES and let $(\sigma^{(g)}, g \in \mathbb{N})$ and $(\mathbf{C}^{(g)}, g \in \mathbb{N})$ be respectively the sequences of step-sizes and covariance matrices. Assume that $\mathbf{m}^{(g)}, a\sigma^{(g)}, \mathbf{C}^{(g)}$ are given, the λ new points or *individuals* are sampled in step 1. according to:

$$\mathbf{x}_i^{(g)} = \mathbf{m}^{(g)} + a\sigma^{(g)} \underbrace{\mathcal{N}_i(0, \mathbf{C}^{(g)})}_{=\mathbf{y}_i}, \quad \text{for } i = 1, \cdots, \lambda \ . \quad (5)$$

Those λ individuals are evaluated in step 2. and ranked according to f:

$$f(\mathbf{x}_{1:\lambda}^{(g)}) \leq \cdots \leq f(\mathbf{x}_{\mu:\lambda}^{(g)}) \leq \cdots \leq f(\mathbf{x}_{\lambda:\lambda}^{(g)}) \ , \quad (6)$$

where we use the notation $\mathbf{x}_{i:\lambda}^{(g)}$ for i^{th} best individual.

The mean $\mathbf{m}^{(g)}$ is then updated by taking the weighted mean of the best μ individuals:

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} \omega_i \mathbf{x}_{i:\lambda}^{(g)} = \mathbf{m}^{(g)} + a\sigma^{(g)} \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda} \ , \quad (7)$$

where $\mathbf{y}_{i:\lambda} = (\mathbf{x}_{i:\lambda}^{(g)} - \mathbf{m}^{(g)})/\sigma^{(g)}$. In general $\mu = \frac{\lambda}{2}$ and $(\omega_i)_{1\leq i\leq\mu}$ are strictly positive and normalized weights, i.e., satisfying $\sum_{i=1}^{\mu} \omega_i = 1$. This update displaces the mean vector towards the best solutions. The increment $a\sigma^{(g)} \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}$ has an interpretation in terms of (stochastic) approximation of the gradient with respect to $\mathbf{m}$ of a joint criterion $\mathcal{J}$ mapping $(\mathbf{m}, \sigma, \mathbf{C})$ to $\mathbb{R}$ and depending on quantiles of the objective function f [2].

A measure characterizing the recombination used is called *the variance effective selection mass* and defined by $\mu_{\text{eff}} = \left(\sum_{i=1}^{\mu} {\omega_i}^2\right)^{-1}$. The choice of the recombination type has an important impact on the efficiency of the algorithm [1]. The default weights are equal to:

$$\omega_i = \frac{\ln(\mu+1) - \ln(i)}{\mu \ln(\mu+1) - \ln(\mu!)}, \qquad \text{for } i = 1, \cdots, \mu \ . \quad (8)$$

The update of the covariance matrix $\mathbf{C}^{(g)}$ uses two mechanisms. First of all the *rank-one update* [33] using the so called evolution path $\mathbf{p}_{\text{c}}^{(\text{g})} \in \mathbb{R}^n$ whose update is given by:

$$\mathbf{p}_{\text{c}}^{(\text{g}+1)} = (1-c_{\text{c}})\mathbf{p}_{\text{c}}^{(\text{g})} + \sqrt{c_{\text{c}}(2-c_{\text{c}})\mu_{\text{eff}}} \, \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \ , \quad (9)$$

where $c_c \in]0, 1]$. For the constant $c_c = 1$, the evolution path points towards the descent direction $\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}$ and for $c_c \neq 1$, the vector $\mathbf{p}_{\text{c}}^{(\text{g})}$ adds the steps followed by the mean vector over the iterations using some normalization to dampen previous steps, so as

not to rely too much on old information. The vector $\mathbf{p}_{\mathrm{c}}^{(\mathrm{g}+1)}$ gives a direction where we expect to see good solutions. From the evolution path, the rank-one matrix $\mathbf{p}_{\mathrm{c}}^{(\mathrm{g}+1)}\mathbf{p}_{\mathrm{c}}^{(\mathrm{g}+1)^T}$ is built and added to the covariance matrix (see Eq. (10)). Geometrically it deforms the ellipsoid-density in the direction $\mathbf{p}_{\mathrm{c}}^{(\mathrm{g}+1)}$, i.e., the rank-one update increases the probability to sample in the next iteration in the direction $\mathbf{p}_{\mathrm{c}}^{(\mathrm{g}+1)}$.

The second mechanism is the *rank-mu update* [31] where the rank-mu matrix $\sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}\mathbf{y}_{i:\lambda}^{\mathrm{T}}$ is added to the covariance matrix. This rank-mu matrix is also the stochastic approximation of the gradient of the joint criterion $\mathcal{J}$ with respect to $\mathbf{C}$ [2]. The update of the covariance matrix combines rank-one and rank-mu update and reads:

$$\mathbf{C}^{(g+1)} = (1-c_{\mathrm{cov}})\mathbf{C}^{(g)} + \underbrace{\frac{c_{\mathrm{cov}}}{\mu_{\mathrm{cov}}}\mathbf{p}_{\mathrm{c}}^{(\mathrm{g}+1)}\mathbf{p}_{\mathrm{c}}^{(\mathrm{g}+1)^{\mathrm{T}}}}_{\text{rank-one update}} + \underbrace{c_{\mathrm{cov}}\left(1-\frac{1}{\mu_{\mathrm{cov}}}\right)\times\sum_{i=1}^{\mu}\omega_i\mathbf{y}_{i:\lambda}\mathbf{y}_{i:\lambda}^{\mathrm{T}}}_{\text{rank-mu update}} \quad . \tag{10}$$

The initial evolution path $\mathbf{p}_{\mathrm{c}}^{(0)}$, c_{c}, c_{cov} and μ_{cov} are parameters of the algorithm. Default values can be found in [30].

In addition to the covariance matrix adaptation, the step-size $a\sigma^{(g)}$ is controlled after every iteration. To perform the adaptation, a conjugate evolution path $\mathbf{p}_{\sigma}^{(\mathrm{g})} \in \mathbb{R}^n$ at generation g is updated according to:

$$\mathbf{p}_{\sigma}^{(\mathrm{g}+1)} = (1-c_{\sigma})\mathbf{p}_{\sigma}^{(\mathrm{g})} + \sqrt{c_{\sigma}(2-c_{\sigma})\mu_{\mathrm{eff}}}\mathbf{C}^{(g)^{-\frac{1}{2}}}\frac{\mathbf{m}^{(g+1)}-\mathbf{m}^{(g)}}{\sigma^{(g)}} \quad . \tag{11}$$

The step-size is adapted according to:

$$\sigma^{(g+1)} = \sigma^{(g)}\exp\left(\frac{c_{\sigma}}{d_{\sigma}}\left(\frac{\|\mathbf{p}_{\sigma}^{(\mathrm{g}+1)}\|}{\mathrm{E}\|\mathcal{N}(0,\mathbf{I})\|}-1\right)\right) \quad , \tag{12}$$

where $\mathbf{p}_{\sigma}^{(0)}$, c_{σ} and d_{σ} are parameters of the algorithm with default values defined in [30]. This update rule implements to increase the step-size when the length of the conjugate evolution path is larger than the length it would have if selection would be random (this length will then be equal to $\|\mathcal{N}(0,\mathbf{I})\|$) and decrease it otherwise.

All the updates rely on the ranking determined by Eq. 6 only and not on the exact value of the objective functions making the algorithm invariant to monotonic transformations of the objective functions that preserve the ranking of solutions.

On the class of functions $\mathbf{x} \mapsto g_{\mathcal{M}} \circ f_{\mathrm{cq}}(\mathbf{x})$ where f_{cq} is a convex quadratic function and $g_{\mathcal{M}} : \mathbb{R} \to \mathbb{R}$ a monotonically increasing function, the covariance matrix sequence $\mathbf{C}^{(g)}$ becomes proportional to the inverse Hessian of the function $f_{\mathrm{cq}}(\mathbf{x})$, i.e., the algorithm is able to learn second order information without using any derivatives.

Step-size adaptation is important to achieve fast convergence corresponding to linear convergence with rates close to optimal rates that can be achieved by evolution strategies algorithms. In combination with covariance matrix adaptation, step-size adaptation allows to achieve linear convergence on a wide range of functions including *ill-conditioned problems*.

3.2 Handling constraints with CMA-ES

Several methods are used, in the literature, to handle constraints in stochastic optimization algorithms. In general, unfeasible individuals can be rejected, penalized or repaired. In the following, we briefly discuss these alternatives. A more detailed study and comparison can be found in [42].

- Rejection of unfeasible individuals: Besides its simplicity and ease of implementation, rejecting the unfeasible individuals, also called "death penalty" does not require any parameter to be tuned. However, ignoring unfeasible individuals can prevent the algorithm from finding the region containing the optimum solution if it is close to the feasible domain boundaries [41];
- Penalizing unfeasible individuals: Penalization is the most widespread approach used to handle constraints. This method corresponds to a transformation of the optimization problem:

$$\begin{cases} \min\ f(\mathbf{x}) \\ \text{s.t. } h_i(\mathbf{x}) \le d_i\ \forall i = 1,\cdots,m \\ \qquad \Rightarrow \min\ f(\mathbf{x}) + \sum_{i=1}^{m} g(h_i(\mathbf{x}) - d_i) \end{cases} , \tag{13}$$

  where m is the number of constraints and $g(.)$ is the penalty function which is non-negative, equal to zero in $\mathbb{R}_-$ and increasing in $\mathbb{R}_+$. In general, $g(.)$ contains parameters to be tuned. These parameters depend on the problem to be optimized. A solution to avoid the difficulty of tuning those parameters consists in using an adaptive penalization which does not require any user specified constant. However, penalizing all unfeasible individuals implies evaluating all unfeasible individuals which can be costly;
- Repairing unfeasible individuals: Another popular solution to handle constraints is to repair each unfeasible individual before evaluating it. An important parameter to be specified is the probability of

replacement of the unfeasible individual by the repaired new feasible individual. Moreover, repairing introduces a new individual in the population which may not obey to the adapted distribution, and hence may hold up the optimization process of CMA-ES.

Knowing the limitations of each of the constraint-handling approaches, the approach used in the present paper is a mixture between two approaches: *adaptive penalization of the marginally unfeasible individuals* and *rejection of only the unfeasible individuals far from the boundaries of the feasible domain*. Using this approach, rejecting only individuals far from the feasible domain does not prevent the algorithm from finding a solution near the feasible domain boundaries, and by using adaptive penalization, the critical penalization coefficients are adapted automatically during the course of the search[2].

A box constraint handling is presented in [32] in which the feasible space is a hypercube defined by lower and upper boundary values for each parameter. In the present paper, this approach is generalized in order to handle feasible spaces defined by lower and upper boundary values for a sum of some of the parameters (e.g., to constrain the length of multilateral wells).

Given an optimization problem with a dimension n, let us suppose we have $m \in \mathbb{N}$ constraints denoted by $S_j,\ \forall j = 1, \cdots, m$. For each constraint S_j, we define $P_j \subset \{1, \cdots, n\}$ such that a vector $\mathbf{x} = (x_i)_{1 \leq i \leq n}$ is feasible with respect to the constraint S_j if:

$$v_{(j,-)} < q_j = \sum_{p \in P_j} x_p < v_{(j,+)} \ , \tag{14}$$

where $v_{(j,-)}$ and $v_{(j,+)}$ are the lower and upper boundaries defining S_j. Constraints are then handled as follows, when evaluating an individual $\mathbf{x}$:
- *Initializing weights*: In the first generation, boundary weights γ_j are initialized to $\gamma_j = 0,\ \forall j = 1, \cdots, m$;
- *Setting weights*: From the second generation upwards, if the distribution mean is unfeasible and weights are not set yet

$$\gamma_j \longleftarrow \frac{2\delta_{\text{fit}}}{\sigma^2 \frac{1}{n} \sum_{i=1}^{n} \mathbf{C}_{ii}}, \qquad \forall j = 1, \cdots, m \ , \tag{15}$$

where δ_{fit} is the median from the last $\frac{20+3n}{\lambda}$ generations of the interquartile range of the unpenalized objective function evaluations and $\mathbf{C}_{ii}$ is the i^{th} diagonal element of the covariance matrix;

- *Increasing weights*: For each constraint S_j, if the distribution mean M_j, i.e., the mean of q_j for the λ individuals of the current generation, is out-of-bounds and the distance from M_j to the feasible domain, i.e., $\max(0, M_j - v_{(j,+)}) + \max(0, v_{(j,-)} - M_j)$ is larger than $\sigma \times \sqrt{\frac{1}{\text{card}(P_j)} \sum_{p \in P_j} \mathbf{C}_{pp}} \times \max(1,\ \frac{\sqrt{n}}{\mu_{\text{eff}}})$ then

$$\gamma_j \longleftarrow \gamma_j \times 1.1^{\max(1, \frac{\mu_{\text{eff}}}{10n})}, \qquad \forall j = 1, \cdots, m \ , \tag{16}$$

where $\text{card}(P_j)$ denotes the cardinality of the set P_j;
- *Evaluating the individual*:

$$f(\mathbf{x}) \longleftarrow f(\mathbf{x}) + \frac{1}{m} \sum_{j=1}^{m} \gamma_j \frac{(q_j^{\text{feas}} - q_j)^2}{\xi_j} \ , \tag{17}$$

where q_j^{feas} is the projection of q_j on the feasible domain and $\xi_j = \exp\left(0.9 \left(\frac{1}{\text{card}(P_j)} \sum_{p \in P_j} \log(\mathbf{C}_{pp}) - \frac{1}{n} \times \sum_{i=1}^{n} \log(\mathbf{C}_{ii})\right)\right)$.

An individual $\mathbf{x}$, in the following, will be rejected and resampled if $|q_j^{\text{feas}} - q_j| > p\% \times |v_{(j,+)} - v_{(j,-)}|$, where $p\%$ is a parameter to be chosen. In all runs presented in this paper, $p\%$ is chosen to be equal to 20%.

4 CMA-ES and a real-coded GA for the well placement problem

The choice of a stochastic optimization method was motivated by the ability of this type of algorithms to deal with non-smooth, non-convex and multi-modal functions. In addition, stochastic optimization does not require any gradients and can be easily parallelized. So far, the most popular stochastic approaches for tackling well placement have been genetic algorithms encoding the real parameters to be optimized as bit-strings. However, it is know in the stochastic algorithm community, that representing real vectors as bit strings leads to poor performance [52]. Recently, a comparison between binary and real representations on a well placement problem in a channelized synthetic reservoir model has been made, showing that the continuous variant outperforms the binary one [13].

This section compares a real-coded GA with CMA-ES on a well placement problem. To allow a fair comparison, both algorithms are used without parameter tuning. Indeed, tuning an algorithm requires some extra objective function evaluations that would need to be taken into account otherwise. Default parameters

[2] The penalization method depends in general on other parameters which are on the other hand much less critical and which are tuned beforehand to be suitable for a wide range of problems [32].

are used for the CMA-ES algorithm[3] and typical parameter value for the GA.

4.1 Well placement using CMA-ES

The initial population is normally drawn using a mean vector uniformly drawn in the reservoir. Parameters were defined according to default settings [30].

The population size λ is an important parameter of CMA-ES [30]. The default population size value equals $4 + \lfloor 3 \times \ln(D) \rfloor$, where D is the dimension of the problem. Independent restarts with increasing population size are suggested in [4]. In this paper, the optimal tuning of the population size was not addressed. However, due to the difficulty of the problem at hand, we use a population size greater than the default value.

4.2 Well placement using GA

Genetic algorithms [35,36] are stochastic search algorithms that borrow some concepts from nature. Similar to CMA-ES, GAs are based on an initial population of individuals. Each individual represents a possible solution to the problem at hand. Starting with an initial population of points called individuals or chromosomes, and at each iteration, candidate solutions evolve by selection, mutation and recombination until reaching the stopping criteria with a satisfactory solution. The correspondence between a solution and its representation needs to be defined. In general, simple forms like an array or a matrix of integer or bit elements are used. In this paper, individuals are parameterized as defined for CMA-ES (see Section 2.2). Hence, well coordinates are defined using a real encoding. Elitism is used to make sure that the best chromosome would survive to the next generation. The used operators are defined as follows:

- The crossover starts with two parent chromosomes causing them to unite in points to create two new elements. The greater chromosome fitness' rank, the higher probability it will be selected. After selecting the two parents, crossover is applied with a probability denoted crossprob. To apply the crossover, we randomly draw an index i between 1 and D and a number c between 0 and 1. Let us denote the two parents by $(x_{1,j})_{1 \leq j \leq D}$ and $(x_{2,j})_{1 \leq j \leq D}$, then $x_{1,i} \leftarrow c \times x_{1,i} + (1-c) \times x_{2,i}$ and $x_{2,i} \leftarrow c \times x_{2,i} + (1-c) \times x_{1,i}$.

[3] At the exception of the population size where the default setting is known to be good for non-rugged landscapes but needs to be increased otherwise [30]

Table 2 GA parameters: the probabilities to apply GA operators, i.e., crossover and mutation.

Constant	Value
crossprob	0.7
mutprob	0.1

- The mutation, instead, starts with one individual and randomly changes some of its components. Mutation is applied to all chromosomes, except the one with the best fitness value, with a probability of mutation denoted mutprob. In this case, we randomly draw an index i. Let us denote the selected chromosome by $(x_j)_{1 \leq j \leq D}$, then $x_i \leftarrow \min_i + c \times (\max_i - \min_i)$, where $\min_i$ and $\max_i$ are the minimum and the maximum values that can be taken by the i^{th} coordinate of the chromosome and c is a number randomly drawn between 0 and 1.

The mutation and crossover probabilities are set to typical values (see Table 2)[4].

To handle the constraints, the genetic algorithm is combined with the Genocop III technique (Genetic Algorithm for Numerical Optimization of Constrained Problems) [18]. This procedure maintains two separate populations. The first population called the search population contains individuals which can be unfeasible. The second population, the reference population, consists of individuals satisfying all constraints (linear and non-linear), called reference individuals. Feasible search individuals and reference individuals are evaluated directly using the objective function. However, unfeasible individuals are repaired before being evaluated. More details about Genocop III can be found in [43].

4.3 Well placement performance

All tests performed in the present work are conducted on the PUNQ-S3 test case [21]. PUNQ-S3 is a case taken from a reservoir engineering study on a real field. The model contains $19 \times 28 \times 5$ grid blocks. We suppose that the field does not contain any production or injection well initially. The elevation of the field and its geometry is shown in Fig. 3. We plan to drill two wells: one unilateral injector and one unilateral producer. The dimension of the problem is then equal to $12 (= 6 \times 2)$.

To compare results obtained by both CMA-ES and the genetic algorithm, 14 runs were performed for each algorithm. A streamline simulator is used during the

[4] A good choice of the crossover probability is said to be in between 0.4 and 0.9 [54,14], 0.6 and 0.8 [27], 0.6 and 0.95 [20, 22], 0.6 and 0.8 [51]. A good choice of the mutation probability is said to be in between 0.001 and 0.1 [14,20,22], 0.005 and 0.05 [51], 0.05 and 0.1 [54].

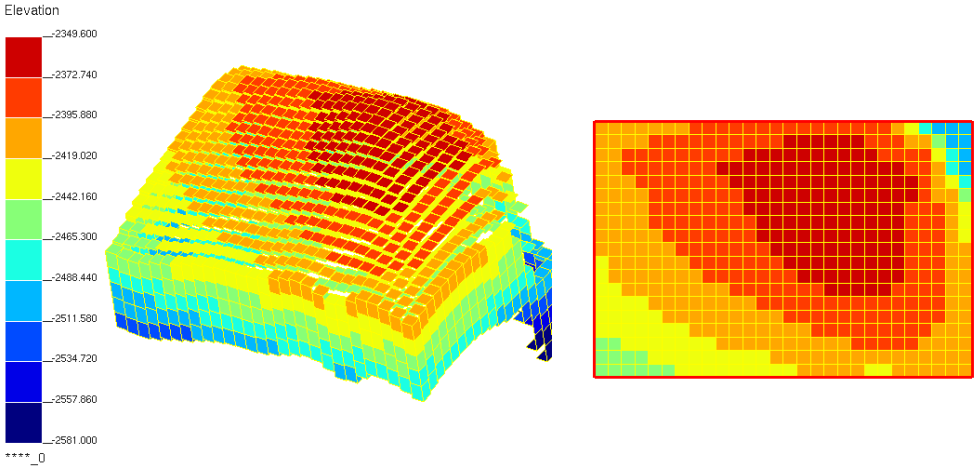

Fig. 3 Elevation (in meters) and geometry of the PUNQ-S3 test case.

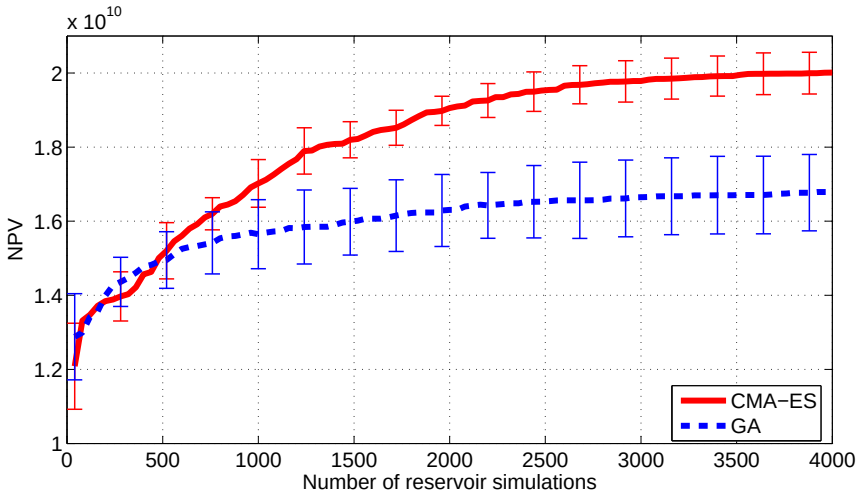

Fig. 4 The mean value of NPV (in US dollars) and its corresponding standard deviation for well placement optimization using CMA-ES (*solid line*) and GA (*dashed line*). Fourteen runs are performed for each algorithm. Constraints are handled using an adaptive penalization with rejection technique for CMA-ES and using Genocop III for GA.

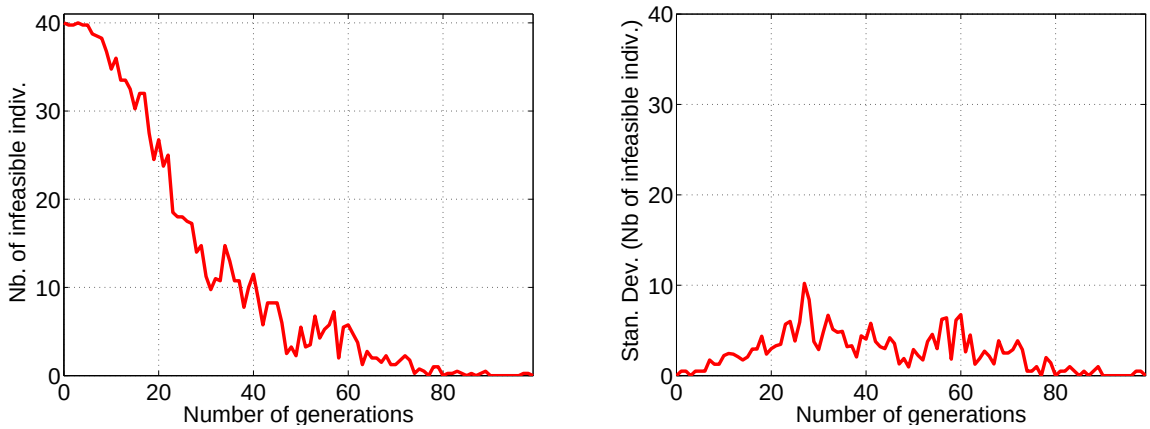

Fig. 5 The mean number of unfeasible individuals per generation and its corresponding standard deviation using CMA-ES with an adaptive penalization with rejection technique. Here, we consider only unfeasible individuals far from the feasible domain, i.e., resampled individuals.

optimization. In this comparison, a bottomhole pressure imposed on the producer is fixed to 80 bar, and a bottomhole pressure imposed on the injector is fixed to 6.000 bar which is too high. This unrealistic value was used only for the sake of comparison between the two optimization methods.

The population size is set to 40 for both algorithms. The stopping criterion is also the same for both of the methods: a maximum number of iterations equal to 100. The size of the reference population for Genocop III is set to 60. Well lengths are constrained with a maximum well length $L_{\max} = 1000$ meters.

Fig. 4 shows the average performance and its standard deviation of the well placement optimization using both algorithms measured by the overall best objective function value. It is clear that CMA-ES outperforms the GA: the genetic algorithm adds only 40% to the best NPV obtained by a randomly sampled configuration, i.e., in the first generation of the optimization. However, CMA-ES adds 80%.

Fig. 5 shows that CMA-ES handles the used constraints successfully. The number of well configurations resampled, i.e., far from the feasible domain, approaches to 0 at the end of the optimization. Fig. 5 shows that after a number of iterations, the majority of the well configurations generated by CMA-ES are either feasible or close to the feasible domain.

Fig. 6 shows the positions of "optimum" wells obtained from 14 runs using CMA-ES. CMA-ES succeeds in defining in 11 runs of the 14 performed the same potential zone to place the producer and the injector. This region gives an NPV between $\$1.99 \times 10^{10}$ and $\$2.05 \times 10^{10}$. In the other three runs, CMA-ES finds each time a different local optimum with NPV values equal to: $\$1.83 \times 10^{10}$, $\$1.95 \times 10^{10}$ and $\$2.05 \times 10^{10}$. Despite the large number of local optima, CMA-ES succeeds in providing satisfactory results on 93 % of the runs, if we consider that a run is satisfactory if it gives an NPV greater or equal to $\$1.95 \times 10^{10}$.

For the genetic algorithm, 14 runs were performed to trace different "optimum" well configurations in Fig. 7. Well configurations are not concentrated in some well-defined regions and have an NPV mean value equal to $\$1.68 \times 10^{10}$ with a standard deviation equal to 1.06×10^9. The GA leads to well configurations dispersed over a large zone. The maximum value of NPV obtained by the GA is equal to $\$1.86 \times 10^{10}$ and it corresponds to a well configuration close to a well configuration obtained by CMA-ES with an NPV $\$2.05 \times 10^{10}$.

Results confirm that CMA-ES is able to find in the majority of the runs a solution in the same potential region. In 93% of the runs on the considered test case, CMA-ES finds a well configuration with a satisfactory

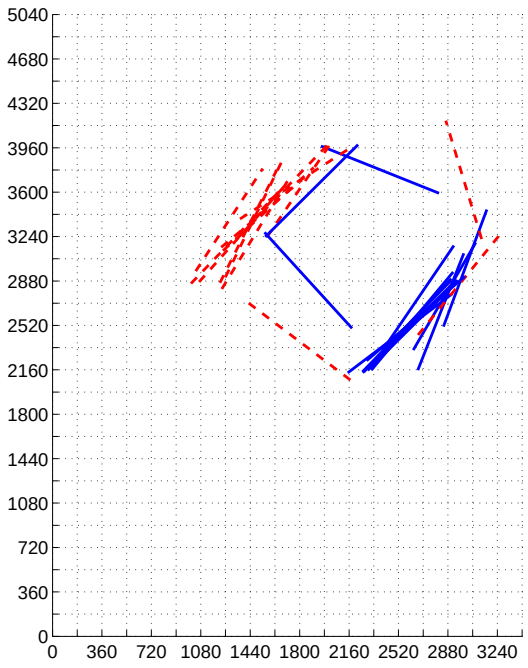

Fig. 6 The positions of solution wells found by 14 runs of CMA-ES projected on the top face of the reservoir. Injectors are represented by (*dashed line*). Producers are represented by (*solid line*).

NPV value. However, the GA has difficulties to define this potential region and seems to prematurely converge in different regions. Premature convergence in the GA is most certainly due to the lack of mechanisms that (1) would play the role of the step-size mechanism in CMA-ES which is able to increase the step-size in linear environments and (2) would play the role of the covariance matrix adaptation mechanism allowing to adapt the main search directions (elongate / shrink certain directions and learn the principal axis of the problem) to solve efficiently ill-conditioned problems. Without this latter mechanism on ill-conditioned problems, it is common to observe premature convergence.

5 Meta-models for CMA-ES

The CMA-ES algorithm makes minimal assumptions on the objective function by exploiting *solely* the ranking of solutions. However, it can be reasonable to assume that locally, the objective function can be approximated by its second order approximation and to try to exploit this within the algorithm to reduce the number of (expensive) reservoir simulations. The use of local meta-models instead of a global one is motivated by the fact

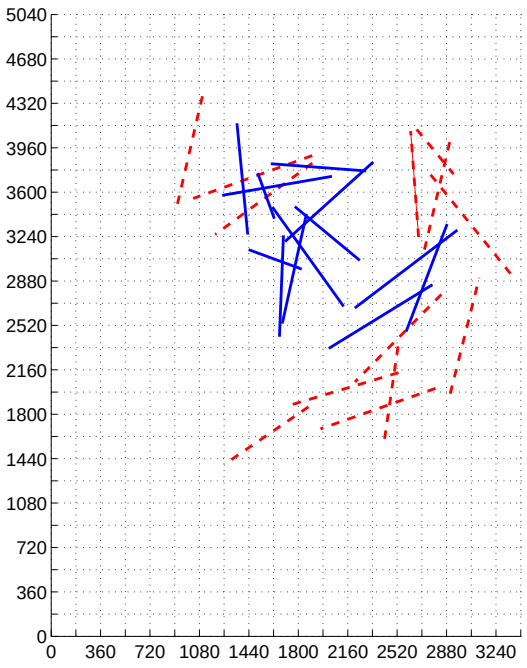

Fig. 7 The positions of solution wells found by 14 runs of the GA projected on the top face of the reservoir. Injectors are represented by (*dashed line*). Producers are represented by (*solid line*).

that we want the algorithm to be able to handle multimodal functions or unimodal functions where a global quadratic model would model poorly the function. We thus propose now to combine CMA-ES with local meta-models where for each individual in the population, an approximate convex quadratic model is built using true objective function evaluations collected during the optimization process [37].

5.1 Locally weighted regression

To build an approximate model of the objective function f, denoted by $\hat{f}$, we use a locally weighted regression. During the optimization process, a database, i.e., a training set is built by storing, after every evaluation on the true objective function, points together with their objective function values $(\mathbf{x}, y = f(\mathbf{x}))$. Assuming that the training set contains a sufficient number m of couples $(\mathbf{x}, f(\mathbf{x}))$, let us consider an individual denoted $\mathbf{q} \in \mathbb{R}^n$ to be evaluated with the approximate model, where n is the dimension of the problem. We begin by selecting the k nearest points $(\mathbf{x}_j)_{1 \leq j \leq k}$ from the training set. The distance used for this purpose exploits the natural metric defined by the covariance matrix of CMA, namely the Mahalanobis distance with respect to the current covariance matrix $\mathbf{C}$ defined for two given points $\mathbf{z}_1 \in \mathbb{R}^n$ and $\mathbf{z}_2 \in \mathbb{R}^n$ by $d_{\mathbf{C}}(\mathbf{z}_1, \mathbf{z}_2) = \sqrt{(\mathbf{z}_1 - \mathbf{z}_2)^T \mathbf{C}^{-1} (\mathbf{z}_1 - \mathbf{z}_2)}$. We build with locally weighted regression an approximate objective function using (true) evaluations $(y_j)_{1 \leq j \leq k}$ corresponding to the k selected nearest points to $\mathbf{q}$.

The use of a full quadratic meta-model is suggested in [38]. Hence, using a vector $\beta \in \mathbb{R}^{\frac{n(n+3)}{2}+1}$, we define $\hat{f}$ as follows:

$$\hat{f}(\mathbf{x}, \beta) = \beta^T \left(x_1^2, \cdots, x_n^2, \cdots, x_1 x_2, \cdots, x_{n-1} x_n, x_1, \cdots, x_n, 1\right)^T . \quad (18)$$

The full quadratic meta-model is built based on minimizing the following criterion with respect to the vector of parameters β of the meta-model at $\mathbf{q}$:

$$A(\mathbf{q}) = \sum_{j=1}^{k} \left[\left(\hat{f}(\mathbf{x}_j, \beta) - y_j \right)^2 K\left(\frac{d_{\mathbf{C}}(\mathbf{x}_j, \mathbf{q})}{h} \right) \right] . \quad (19)$$

The kernel weighting function $K(.)$ is defined by $K(\zeta) = (1 - \zeta^2)^2$, and h is the bandwidth defined by the distance of the k^{th} nearest neighbor data point to $\mathbf{q}$ where k must be greater or equal to $\frac{n(n+3)}{2} + 1$ for a full quadratic meta-model.

5.2 Approximate ranking procedure

To incorporate the approximate model built using the locally weighted regression, we use the approximate ranking procedure [49]. This procedure decides whether the quality of the model is good enough in order to continue exploiting this model or new simulations should be performed. The resulting method is called the local-meta-model CMA-ES (lmm-CMA) [38]. A new variant called the new-local-meta-model CMA-ES (nlmm-CMA) was proposed in [9] improving over lmm-CMA on most benchmark test functions.

In this paper, we use the variant nlmm-CMA$_2$ defined in [9]. For a given generation, let us denote individuals of the current population of CMA-ES by $(\mathbf{x}_i)_{1 \leq i \leq \lambda}$, where λ is the population size. The following procedure is then performed:

1. *build* $\hat{f}(\mathbf{x}_i)$ for all individuals of the current population $(\mathbf{x}_i)_{1 \leq i \leq \lambda}$.
2. *rank* individuals according to their approximated value $\hat{f}(\mathbf{x}_i)$ and *determine* the μ best individuals set and the best individual.
3. *evaluate* the best individual with the true objective function and *add* its evaluation to the training set.
4. for n_{ic} from 1 to $(\lambda - 1)$, we:
   (a) *build* $\hat{f}(\mathbf{x}_i)_{1 \leq i \leq \lambda}$.

(b) *rank* individuals according to their approximated value $\hat{f}(\mathbf{x}_i)$[5] and *determine* the μ best individuals set and the best individual.
(c) if less than one fourth of the population is evaluated, the meta-model is accepted if it succeeds in keeping both **the best individual** and **the ensemble of μ best individuals** unchanged.
(d) if more than one fourth of the population is evaluated, the meta-model is accepted if it succeeds in keeping **the best individual** unchanged.
(e) if the meta-model is accepted, we break. If not, we *evaluate* the best unevaluated individual with the true objective function, *add* its evaluation to the training set, and *loop* to step 4, until reaching the acceptance criterion of the meta-model.

Hence, $(1+n_{ic})$ individuals are evaluated for a given generation where n_{ic} is the number of iteration cycles needed to satisfy the meta-model acceptance criterion.

6 Application of CMA-ES with meta-models on the PUNQ-S3 case

The proposed CMA-ES with meta-models is able to handle different possible well configurations as defined in Section 2.2. In the following, the performance of the approach is demonstrated only on two cases.

6.1 Placement of one unilateral producer and one unilateral injector

In this application, we consider a placement problem of one unilateral injector and one unilateral producer on the PUNQ-S3 case. Parameters of the problem are the same as for the example in Section 4.3, except for the following differences:

- a commercial reservoir simulator is used to evaluate field productions of each phase instead of the streamline simulator;
- the bottomhole pressure imposed on the producer is fixed to 150 bar;
- the bottomhole pressure imposed on the injector is fixed to 320 bar.

To define the parameters of the meta-model, we choose k, the number of individuals used to evaluate the meta-model, equal to 100. Meta-models are used when the training set contains at least 160 couples of points with their evaluations. For each method, i.e., CMA-ES and CMA-ES with local meta-models (lmm-CMA), 10 runs were performed. The evolution of the NPV mean value in term of the mean number of reservoir simulations is represented in Fig. 8.

[5] Or true objective function if the individuals have been evaluated on it.

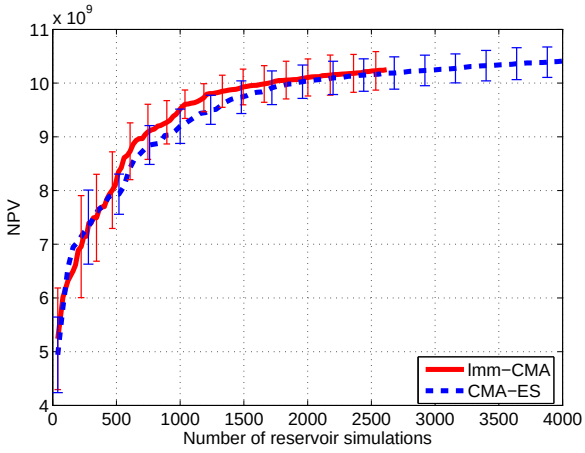

Fig. 8 The mean value of NPV (in US dollars) and its corresponding standard deviation for well placement optimization using CMA-ES with meta-models (*solid line*) and CMA-ES (*dashed line*). Ten runs are performed for each algorithm. Constraints are handled using an adaptive penalization with rejection technique.

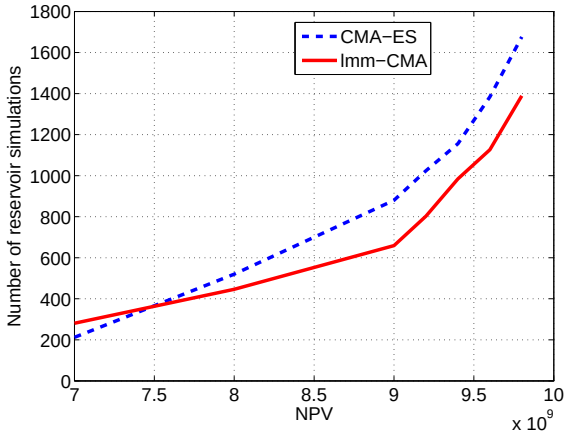

Fig. 9 The mean number of reservoir simulations needed to reach a given NPV value using CMA-ES with meta-models (*solid line*) and CMA-ES (*dashed line*). Ten runs are performed for each algorithm.

Fig. 8 shows that, for the same number of reservoir simulations, combining CMA-ES with meta-models allows to reach higher NPV values compared to CMA-ES, given a restricted budget of reservoir simulations. A better representation is to show the mean number of reservoir simulations needed to reach a certain value of NPV for CMA-ES and for CMA-ES with meta-models (Fig. 9). To reach an NPV value of $\$9 \times 10^9$, lmm-CMA requires only 659 reservoir simulations, while CMA-ES requires 880 reservoir simulations. If we consider that an NPV equal to $\$9 \times 10^9$ is satisfactory, using meta-models reduces the number of reservoir simulations by

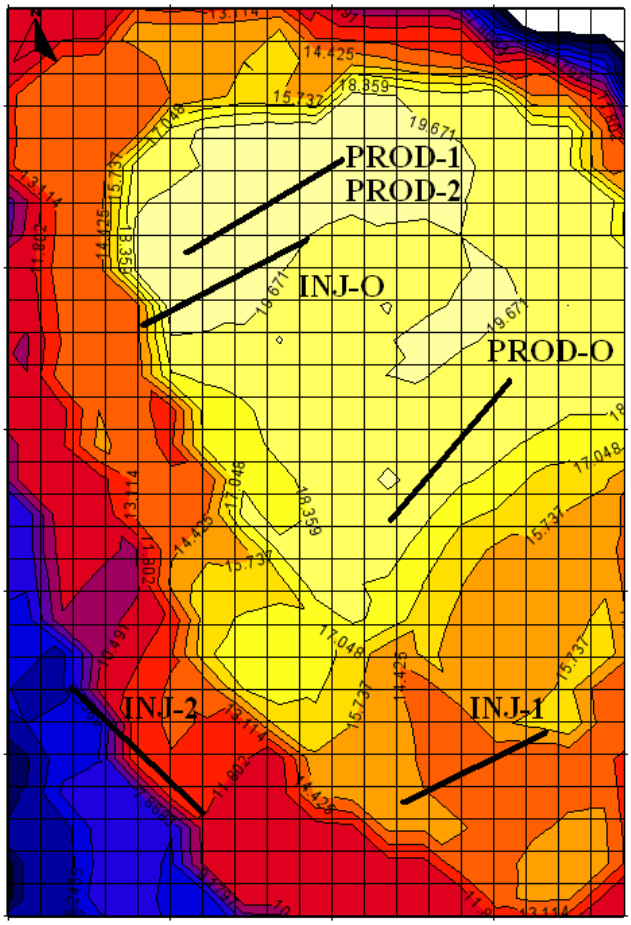

Fig. 10 The SoPhiH Map, with solution well configuration obtained using CMA-ES with meta-models (PROD-O, INJ-O) and two engineer's proposed well configurations (PROD-1, INJ-1 and PROD-2, INJ-2).

25%. For an NPV value equal to $\$9.6 \times 10^9$, the use of meta-models reduces the number of reservoir simulations by 19%. Figs. 8 and 9 highlight the contribution of meta-models in reducing the number of reservoir simulations. Results show also that, in addition to reducing the number of objective function evaluations, the method still succeeds in reaching high NPV values and results are similar to those obtained by CMA-ES. As for the example in Section 4.3, the well placement optimization still succeeds in identifying in the majority of the runs the same potential region to contain optimum wells. In this paper, we present detailed results obtained only by one of the solution well configurations proposed by lmm-CMA. The selected solution well configuration is denoted *optimized config* in the sequel. *Optimized config* is then compared to two configurations designed after some trials in a way to represent the decision of a reservoir engineer (denoted *config.1* and *config.2*). The locations and trajectories of the considered well configurations are shown in Fig. 10.

The engineer's proposed configurations were defined according to the SoPhiH map (Fig. 10) which represents the distribution of the hydrocarbon pore volume over the n_{layers} layers defined by $\sum_{k=1}^{n_{\text{layers}}} (\text{H}_k \times \phi \times S_o)$, where H_k is the gross thickness of the layer k, S_o is the oil saturation and ϕ is the porosity. PROD-c and INJ-c denote respectively the producer and the injector corresponding to the well configuration c. The well configuration is either *config.1*, *config.2* or *optimized config* denoted respectively 1, 2, O. Engineer's proposed wells are horizontal wells where producers (PROD-1 = PROD-2) are placed in the top layer ($k = 1$) and injectors in the bottom layer ($k = 5$). However, producers and injectors in *optimized config* are inclined wells placed in the layer ($k = 3$). The engineer's proposed producer is placed in the region with the highest SoPhiH value.

Fig. 11 shows the production curves of the considered well configurations. The cumulative oil production for *optimized config*, during the 11 simulated years equals 205 MMbbl. However, *config.1* offers only 119 MMbbl and *config.2* offers 102 MMbbl. Therefore, the optimization methodology adds 72% to the best considered engineer's proposed well configuration. *Optimized config* offers also the smallest water cut (0.45 for *optimized config*, 0.57 for *config.1* and 0.69 for *config.2*).

6.2 Placement of one multi-segment producer

In this application, we consider a placement problem of one multi-segment well on the PUNQ-S3 case. We suppose that an injector is already placed in the reservoir. It corresponds to the well denoted INJ-O in Fig. 10. We plan to drill a multi-segment well with two completed segments. The dimension of the problem is then equal to $9 (= 6 + 3)$. The different parameters of the problem are the same as in the example in Section 6.1, except for the population size which is equal to 30. Ten runs were performed with a maximum number of iterations equal to 100.

Fig. 12 shows the evolution of the average performance of the well placement, i.e., NPV mean values and the corresponding standard deviation. Optimizing the placement of one multi-segment producer offers an NPV equal to $\$1.10 \times 10^9 \pm 4.37 \times 10^7$. To reach an NPV mean value of $\$1.10 \times 10^9$, the optimization process requires only 504 reservoir simulations.

The positions of solution wells are shown in Figs. 13 and 14. In this application, the used methodology succeeds in reaching NPV values greater than $\$1.09 \times 10^9$ and in defining an "optimum" well configuration in the same potential region for all the performed runs. Therefore, performing only one run can be conclusive and can

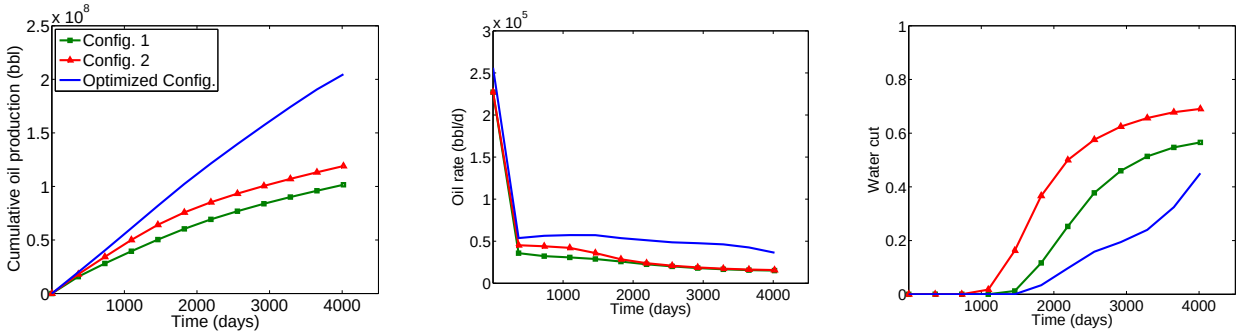

Fig. 11 Production curves for an optimized solution using CMA-ES with meta-models (*optimized config.*) and two engineer's proposed configurations (*config.1* and *config.2*).

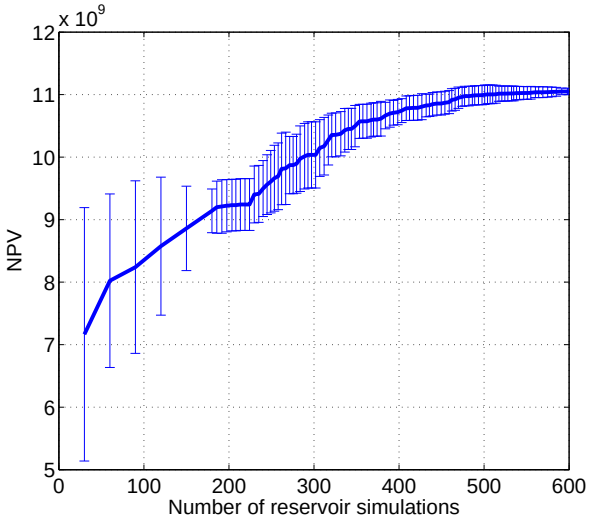

Fig. 12 The mean value of NPV (in US dollars) and its corresponding standard deviation for well placement optimization using CMA-ES with meta-models of one multi-segment well. Ten runs are performed.

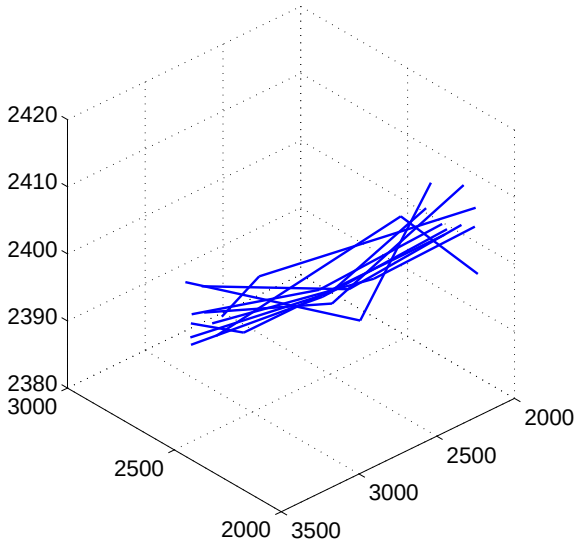

Fig. 13 The positions of solution multi-segment producers found by 10 runs of CMA-ES with meta-models. A zoom on the region containing the solution wells is performed.

ensure converging to a solution well with a satisfactory NPV.

7 Conclusions

In this paper, the stochastic optimization method CMA-ES was applied on a well placement problem. A technique based on adaptive penalization with rejection was developed to handle well placement constraints with CMA-ES. Results showed that this technique ensures that after a number of iterations, the majority of well configurations generated by CMA-ES are either feasible or close to the feasible domain. The optimization with CMA-ES was compared to a GA which is the most popular method used in well placement optimization in the literature. Both algorithms were used without parameter tuning allowing for a direct fair comparison of the results. Indeed parameter tuning requires extra function evaluations that should be taken into account when presenting comparison results. In addition, we think that parameter tuning should be done by the designer of the algorithm and not the user as it is unrealistic to waste expensive function evaluations for correcting the weakness of the design phase. The CMA-ES example shows that providing parameter-free algorithms with robust setting is possible to achieve. CMA-ES was shown to outperform the genetic algorithm on the PUNQ-S3 case by leading to a higher net present value (NPV). Moreover, CMA-ES was shown to be able to define potential regions containing optimal well configurations. On the other hand, the genetic algorithm converged to solutions located in different regions for every performed run. In addition those solutions are associated to much smaller NPV values than the solu-

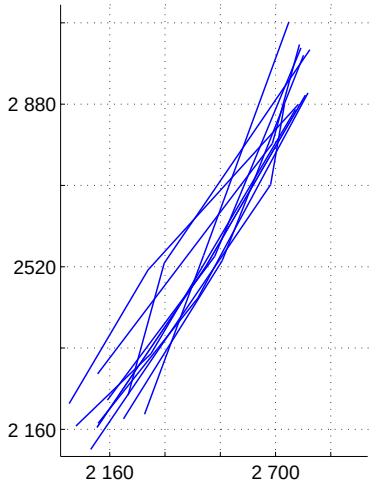

Fig. 14 The positions of solution multi-segment producers found by 10 runs of CMA-ES with meta-models projected on the top face of the reservoir. A zoom on the region containing the solution wells is performed.

tions found by CMA-ES. On the PUNQ-S3 case, the mean NPV value found by GA is 1.68×10^{10}\$. However, the mean NPV value found by CMA-ES is 2.01×10^{10}\$. The ability of CMA-ES to find much higher NPV values and to converge to the same region of the search space, has been explained by its advanced adaptation mechanism that allows the algorithm, on ill-conditioned non-separable problems, to adapt in an efficient way its sampling probability distribution.

To tackle the computational issue related to the number of reservoir simulations performed during the optimization, the proposed methodology was coupled with local-meta-models, and then demonstrated on the PUNQ-S3 case. The use of meta-models was shown to offer similar results (solution well configurations and the corresponding NPV values) as CMA-ES without meta-models and moreover to reduce the number of simulations by 19-25% to reach a satisfactory NPV. The comparison of the obtained results with some engineer's proposed well configurations showed that the proposed optimization methodology is able to provide better well configurations in regions that might be difficult to determine by reservoir engineers. Though the gain brought by coupling meta-models to CMA-ES can seem rather small, we think that it is mainly due to the fact that CMA-ES is almost optimal [29] and thus hard to improve by an important factor. We believe however that performance over CMA-ES could be more significantly improved by exploiting, within the algorithm, knowledge and relevant information about the optimization problem at hand, such as the problem structure. Some first steps in that direction have been conduced in [10,11] where the fact that the objective function can be split into local components referring to each of the wells where each depends on a smaller number of parameters (i.e., partial separability of the objective function) is exploited. An other approach could be to exploit some a priori information such as well allocation factors and connectivity using the work developed in [15].

This work is rather preliminary in terms of type of meta-models tested as we have only focused on local quadratic approximation. However other types of meta-models could be used like kriging and radial basis functions as we have no a priori that quadratic meta-models are the best models to use for practical purposes. This work has demonstrated the potential huge benefit of applying the CMA-ES methodology over more established stochastic techniques for reservoir applications. We believe that many optimization problems in Geosciences could as well be successfully handled with CMA-ES.

Acknowledgments: We would like to thank the anonymous reviewers for their valuable comments on the first version of the manuscript. We would like to thank Nikolaus Hansen for his insightful comments on this work. This work was partially funded by the ANR-08-COSI-007-12 grant and by the FUI of System@tic Paris-Region ICT cluster through contract DGT 117 407 Complex Systems Design Lab (CSDL).

References

1. Arnold, D.V.: Optimal weighted recombination. In: A.H. Wright, M.D. Vose, K.A. De Jong, L.M. Schmitt (eds.) Foundations of Genetic Algorithms, *Lecture Notes in Computer Science*, vol. 3469, pp. 215–237. Springer Berlin / Heidelberg (2005)
2. Arnold, L., Auger, A., Hansen, N., Ollivier, Y.: Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. ArXiv e-prints (2011)
3. Artus, V., Durlofsky, L.J., Onwunalu, J., Aziz, K.: Optimization of nonconventional wells under uncertainty using statistical proxies. Computational Geosciences **10**, 389–404 (2006)
4. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1769–1776. IEEE Press (2005)
5. Bangerth, W., Klie, H., Matossian, V., Parashar, M., Wheeler, M.F.: An autonomic reservoir framework for the stochastic optimization of well placement. Cluster Computing **8**(4), 255–269 (2005)
6. Bangerth, W., Klie, H., Wheeler, M.F., Stoffa, P.L., Sen, M.K.: On optimization algorithms for the reservoir oil well placement problem. Computational Geosciences **10**(3), 303–319 (2006)
7. Bayer, P., Finkel, M.: Evolutionary algorithms for the optimization of advective control of contaminated aquifer zones. Water Resources Research **40**(6) (2004)

8. Beckner, B.L., Song, X.: Field development planning using simulated annealing - optimal economic well scheduling and placement. In: SPE annual technical conference and exhibition, SPE 30650 (1995)
9. Bouzarkouna, Z., Auger, A., Ding, D.Y.: Investigating the local-meta-model CMA-ES for large population sizes. In: C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. Esparcia-Alcazar, C.K. Goh, J. Merelo, F. Neri, M. Preuß, J. Togelius, G. Yannakakis (eds.) Applications of Evolutionary Computation, *Lecture Notes in Computer Science*, vol. 6024, pp. 402–411. Springer Berlin / Heidelberg (2010)
10. Bouzarkouna, Z., Auger, A., Ding, D.Y.: Local-meta-model CMA-ES for partially separable functions. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation, GECCO '11, pp. 869–876. ACM, New York, NY, USA (2011)
11. Bouzarkouna, Z., Ding, D.Y., Auger, A.: Partially separated meta-models with evolution strategies for well placement optimization. In: SPE EUROPEC/EAGE Annual Conference and Exhibition, SPE 143292 (2011)
12. Bruyelle, J., Lange, A.: Automated characterization of fracture conductivities from well tests. In: SPE EUROPEC/EAGE annual conference and exhibition, SPE 121172 (2009)
13. Bukhamsin, A.Y., Farshi, M.M., Aziz, K.: Optimization of multilateral well design and location in a real field using a continuous genetic algorithm. In: SPE/DGS Saudi Arabia Section Technical Symposium and Exhibition, SPE 136944 (2010)
14. Cordoba, A., Vilar, R., Lavrov, A., Utkin, A.B., Fernandes, A.: Multi-objective optimisation of lidar parameters for forest-fire detection on the basis of a genetic algorithm. Optics & Laser Technology **36**(5), 393 – 400 (2004)
15. da Cruz, P.S., Horne, R.N., Deutsch, C.V.: The quality map: A tool for reservoir uncertainty quantification and decision making. SPE Reservoir Evaluation & Engineering **7**(1), 6–14 (2004)
16. Damp, L., Gonzalez., L.F.: Optimisation of the nose of a hypersonic vehicle using dsmc simulation and evolutionary optimisation. In: 5th AIAA ASSC Space Conference (2005)
17. Ding, D.Y.: Optimization of well placement using evolutionary algorithms. In: SPE EUROPEC/EAGE annual conference and exhibition, SPE 113525 (2008)
18. Emerick, A.A., Silva, E., Messer, B., Almeida, L.F., Szwarcman, D., Pacheco, M.A.C., Vellasco, M.M.B.R.: Well placement optimization using a genetic algorithm with nonlinear constraints. In: SPE reservoir simulation symposium, SPE 118808 (2009)
19. Farmer, C.L., Fowkes, J.M., Gould, N.I.M.: Optimal well placement. In: 12th European Conference on the Mathematics of Oil Recovery (ECMOR XII). EAGE (2010)
20. Ferentinos, K.P., Arvanitis, K.G., Sigrimis, N.: Heuristic optimization methods for motion planning of autonomous agricultural vehicles. J. of Global Optimization **23**, 155–170 (2002)
21. Floris, F.J.T., Bush, M.D., Cuypers, M., Roggero, F., Syversveen, A.R.: Methods for quantifying the uncertainty of production forecasts: A comparative study. Petroleum Geoscience **7**, 87–96 (2001)
22. Fogel, D.B.: An introduction to simulated evolutionary optimization. IEEE Transactions on Neural Networks **5**, 3–14 (1994)
23. Forouzanfar, F., Li, G., Reynolds, C.: A two-stage well placement optimization method based on adjoint gradient. In: SPE annual technical conference and exhibition, SPE 135304 (2010)
24. Guyaguler, B., Horne, R.N.: Optimization of well placement. Journal of Energy Resources Technology **122**(2), 64–70 (2000)
25. Guyaguler, B., Horne, R.N.: Uncertainty assessment of well placement optimization. In: SPE annual technical conference and exhibition, SPE 87663 (2001)
26. Guyaguler, B., Horne, R.N., Rogers, L., Rosenzweig, J.J.: Optimization of well placement in a gulf of mexico waterflooding project. In: SPE annual technical conference and exhibition, SPE 63221 (2000)
27. Hajela, P., Lee, E.: Genetic algorithms in truss topological optimization. International Journal of Solids and Structures **32**(22), 3341 – 3357 (1995)
28. Handels, M., Zandvliet, M.J., Brouwer, D.R., Jansen, J.D.: Adjoint-based well-placement optimization under production constraints. In: SPE reservoir simulation symposium, SPE 105797 (2007)
29. Hansen, N., Auger, A., R. Ros S. Finck, P.P.: Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In: GECCO '10: Proceedings of the 12th annual conference comp on Genetic and evolutionary computation, pp. 1689–1696. ACM, New York, NY, USA (2010)
30. Hansen, N., Kern, S.: Evaluating the CMA evolution strategy on multimodal test functions. In: X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J. Rowe, P. Tino, A. Kabán, H.P. Schwefel (eds.) Parallel Problem Solving from Nature - PPSN VIII, *Lecture Notes in Computer Science*, vol. 3242, pp. 282–291. Springer Berlin / Heidelberg (2004)
31. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation. Evolutionary Computation **11**(1), 1–18 (2003)
32. Hansen, N., Niederberger, A.S.P., Guzzella, L., Koumoutsakos, P.: A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. IEEE Transactions on Evolutionary Computation **13**(1), 180–197 (2009)
33. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**(2), 159–195 (2001)
34. Hill, A.D., Zhu, D., Economides, M.J.: Multilateral Wells. Society of Petroleum Engineers (2008)
35. Holland, J.H.: Outline for a logical theory of adaptive systems. J. ACM **9**(3), 297–314 (1962)
36. Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press (1975)
37. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing **9**(1), 3–12 (2005)
38. Kern, S., Hansen, N., Koumoutsakos, P.: Local meta-models for optimization using evolution strategies. In: T. Runarsson, H.G. Beyer, E. Burke, J. Merelo-Guervós, L. Whitley, X. Yao (eds.) Parallel Problem Solving from Nature - PPSN IX, *Lecture Notes in Computer Science*, vol. 4193, pp. 939–948. Springer Berlin / Heidelberg (2006)
39. Li, C., Heinemann, P.H.: A comparative study of three evolutionary algorithms for surface acoustic wave sensor wavelength selection. Sensors and Actuators B: Chemical **125**(1), 311 – 320 (2007)
40. Mera, N.S.: Passive gamma tomography reconstruction of layered structures in nuclear waste vaults. Inverse Problems **23**(1), 385 – 403 (2007)
41. Michalewicz, Z.: Heuristic methods for evolutionary computation techniques. Journal of Heuristics **1**(2), 177–206 (1996)
42. Michalewicz, Z., Dasgupta, D., Riche, R.G.L., Schoenauer, M.: Evolutionary algorithms for constrained engineering problems. Computers & Industrial Engineering Journal **30**(4), 851–870 (1996)

43. Michalewicz, Z., Nazhiyath, G.: Genocop III: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In: D.B. Fogel (ed.) Proceedings of the Second IEEE International Conference on Evolutionary Computation, pp. 647–651. IEEE Press, Piscataway, NJ, USA (1995)
44. Montes, G., Bartolome, P., Udias, A.L.: The use of genetic algorithms in well placement optimization. In: SPE Latin American and Caribbean Petroleum Engineering Conference, SPE 69439 (2001)
45. Morales, A., Nasrabadi, H., Zhu, D.: A modified genetic algorithm for horizontal well placement optimization in gas condensate reservoirs. In: SPE annual technical conference and exhibition, SPE 135182 (2010)
46. Nakajima, L., Schiozer, D.J.: Horizontal well placement optimization using quality map definition. In: Canadian International Petroleum Conference, 2003-053. Petroleum Society of Canada (2003)
47. Onwunalu, J., Durlofsky, L.J.: Application of a particle swarm optimization algorithm for determining optimum well location and type. Computational Geosciences **14**(1), 183–198 (2010)
48. Pan, Y., Horne, R.N.: Improved methods for multivariate optimization of field development scheduling and well placement design. In: SPE annual technical conference and exhibition, SPE 49055 (1998)
49. Runarsson, T.P.: Constrained evolutionary optimization by approximate ranking and surrogate models. In: X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J. Rowe, P. Tino, A. Kabán, H.P. Schwefel (eds.) Parallel Problem Solving from Nature - PPSN VIII, *Lecture Notes in Computer Science*, vol. 3242, pp. 401–410. Springer Berlin / Heidelberg (2004)
50. Sarma, P., Chen, W.H.: Efficient well placement optimization with gradient-based algorithms and adjoint models. In: SPE intelligent energy conference and exhibition, SPE 112257 (2008)
51. Srinivas, M., Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics **24**, 656–667 (1994)
52. Surry, P.D., Radcliffe, N.J.: Real representations. In: Foundations of Genetic Algorithms 4. Morgan Kaufmann (1996)
53. Vlemmix, S., Joosten, G.J.P., Brouwer, D.R., Jansen, J.D.: Adjoint-based well trajectory optimization in a thin oil rim. In: SPE EUROPEC/EAGE annual conference and exhibition, SPE 121891 (2009)
54. Wrobel, L.C., Miltiadou, P.: Genetic algorithms for inverse cathodic protection problems. Engineering Analysis with Boundary Elements **28**(3), 267 – 277 (2004)
55. Yeten, B., Durlofsky, L.J., Aziz, K.: Optimization of nonconventional well type, location and trajectory. SPE Journal **8**, 200–210 (2003)
56. Zandvliet, M.J., Handels, M., van Essen, G.M., Brouwer, D.R., Jansen, J.D.: Adjoint-based well-placement optimization under production constraints. SPE Journal **13**(4), 392–399 (2008)
57. Zubarev, D.I.: Pros and cons of applying proxy-models as a substitute for full reservoir simulations. In: SPE Annual Technical Conference and Exhibition, SPE 124815 (2009)